Web Server Password Hacking Tools

List of Tools :

1.Hashcat

2. Hydra

3.Netcraft

1.HASHCAT :

Hashcat contains 7 types of attack mode.

A. STRAIGHT MODE :

It has a hash file that contains encrypted form and a text file that has passwords.

It compares and checks for the match of password in text from and a encrypted form .

Cmd for staright mode :

"<u>hashcat -m 0 -a 1 hash.txt cark.txt</u>"

where : -m denotes type of hashcode

-a denotes mode of attack **[0]**

hash.txt denotes file that contains encrypted form

cark.txt denotes file that contains password

Example :

```
File  Actions  Edit  View  Help

┌──(abhijeet㉿vbox)-[~/Desktop/abhi]
└─$ hashcat -m 0 -a 0 hash.txt pass.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================
* Device #1: cpu-haswell-11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz, 4382/8829 MB (2048 MB allocatable), 6MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 2 digests; 2 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB
```

encrypted file

password in text file

Result :

```
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

8a8bb7cd343aa2ad99b7d762030857a2:a1
ed20a959d410ccd843d9e1dfcee04228:a12

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: hash.txt
Time.Started.....: Sun Feb 16 19:50:24 2025 (0 secs)
Time.Estimated...: Sun Feb 16 19:50:24 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (pass.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     1453 H/s (0.04ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 2/2 (100.00%) Digests (total), 2/2 (100.00%) Digests (new)
Progress.........: 2/2 (100.00%)
Rejected.........: 0/2 (0.00%)
Restore.Point....: 0/2 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: a1 → a12
Hardware.Mon.#1..: Util: 13%

Started: Sun Feb 16 19:50:21 2025
Stopped: Sun Feb 16 19:50:26 2025

┌──(abhijeet㉿vbox)-[~/Desktop/abhi]
```

list of password and there encrypted versions

denotes success

B. COMBINATION MODE :

Cmd for staright mode :

“hashcat -m 0 -a 1 hash.txt file1.txt file2.txt”

where :

-m : type of hashcode

-a : mode of attack **[1]**

hash.txt: combined encrypted form of password from file1.txt and file2.txt

file1.txt: Password form file 1

file2.txt:Password from file 2

Example :

```
┌──(abhijeet㉿vbox)-[~/Desktop/abhi]      Folder that contains pass and encrypted file
└─$ hashcat -m 0 -a 1 hash.txt file1.txt file2.txt      Cmd for combination mode
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================
* Device #1: cpu-haswell-11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz, 4382/8829 MB (2048 MB allocatable), 6MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Dictionary cache built:
* Filename..: file1.txt
* Passwords.: 5
* Bytes.....: 13
* Keyspace..: 5
* Runtime...: 0 secs

Dictionary cache built:
* Filename..: file2.txt
* Passwords.: 4
* Bytes.....: 15
* Keyspace..: 4
* Runtime...: 0 secs

Hashes: 4 digests; 4 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Optimizers applied:
```

Result :

```
Approaching final keyspace - workload adjusted.

d76f3d05cc9ac98f1f9160274a39fe33:abhi
e9206237def4b4ef46fd933ed0f5a08f:mohan
6a8ce6560ef2e1bb577447dacc710fa7:savita
13489faf95ad78aec2cbebab40ec5a73:rutu

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: hash.txt
Time.Started.....: Wed Feb 12 12:58:35 2025 (0 secs)
Time.Estimated...: Wed Feb 12 12:58:35 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (file1.txt), Left Side
Guess.Mod........: File (file2.txt), Right Side
Speed.#1.........:    93127 H/s (0.01ms) @ Accel:1024 Loops:4 Thr:1 Vec:8
Recovered........: 4/4 (100.00%) Digests (total), 4/4 (100.00%) Digests (new)
Progress.........: 20/20 (100.00%)
Rejected.........: 0/20 (0.00%)
Restore.Point....: 0/5 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-4 Iteration:0-4
Candidate.Engine.: Device Generator
Candidates.#1....: abhi → tu
Hardware.Mon.#1..: Util: 17%

Started: Wed Feb 12 12:58:34 2025
Stopped: Wed Feb 12 12:58:37 2025
```

cracked if success / Exhausted if failed

specifies the type of hash

target file name

time when started

time when finished

C.BRUTE-FORCE ATTACK MODE :

CMD :

“hashcat -m 0 -a 3 4ab8710d781ba5b13aaf561cafd896b7 ?a?a?a?a?a --increment ”

Where:

-m : type of hashmode

-a : attack mode **[3]**

?a: represents all printable ASCII characters

--increment : checks the password pair by pair and if not found increases the pair count by 1 and so on

Example :

```
┌──(abhijeet㉿vbox)-[~]
└─$ hashcat -m 0 -a 3 293ce69b97641eaa9cee7b1aea2a899f ?a?a?a?a --increment
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================
* Device #1: cpu-haswell-11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz, 4382/8829 MB (2048 MB allocatable), 6MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Brute-Force
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.
```

CMD FOR BRUTE FORCE

MASK

--INCREMENT

ENCRPTED FORM OF PASSWORD

if password is not cracked in first pair then automate it to second pair and so on

Result :

```
File  Actions  Edit  View  Help
Status...........: Exhausted
Hash.Mode........: 0 (MD5)
Hash.Target......: 293ce69b97641eaa9cee7b1aea2a899f
Time.Started.....: Wed Feb 12 13:51:05 2025 (0 secs)
Time.Estimated...: Wed Feb 12 13:51:05 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: ?a?a?a [3]
Guess.Queue......: 3/5 (60.00%)
Speed.#1.........:   125.9 MH/s (2.36ms) @ Accel:1024 Loops:95 Thr:1 Vec:8
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 857375/857375 (100.00%)
Rejected.........: 0/857375 (0.00%)
Restore.Point....: 9025/9025 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-95 Iteration:0-95
Candidate.Engine.: Device Generator
Candidates.#1....: sZq → ~
Hardware.Mon.#1..: Util: 34%

293ce69b97641eaa9cee7b1aea2a899f:a@1#

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 293ce69b97641eaa9cee7b1aea2a899f
Time.Started.....: Wed Feb 12 13:51:05 2025 (1 sec)
Time.Estimated...: Wed Feb 12 13:51:06 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: ?a?a?a?a [4]
Guess.Queue......: 4/5 (80.00%)
Speed.#1.........:   129.3 MH/s (3.60ms) @ Accel:1024 Loops:95 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 59535360/81450625 (73.09%)
Rejected.........: 0/59535360 (0.00%)
Restore.Point....: 620544/857375 (72.38%)
```

exhausted not found passwrod

encrypted to normal text as password is cracked

cracked successful

shows the increment for the pair

4/5 : password was cracked at 4 th pairing out of 5

D:HYBRID WORDLIST + MASK ATTACK MODE :

Hashcat's Hybrid Wordlist + Mask attack mode (-a 6) combines the power of dictionary based attacks with brute-force techniques.

It appends a mask (custom character set) to words from a wordlist to generate potential passwords.

CMD :

“hashcat -m 0 -a 6 6848d756da66e55b42f79c0728e351ad /usr/share/wordlists/rockyou.txt -a 6 ?a?a?a?a?a –increment”

Where :

-m : type of hashcode

-a : attack mode

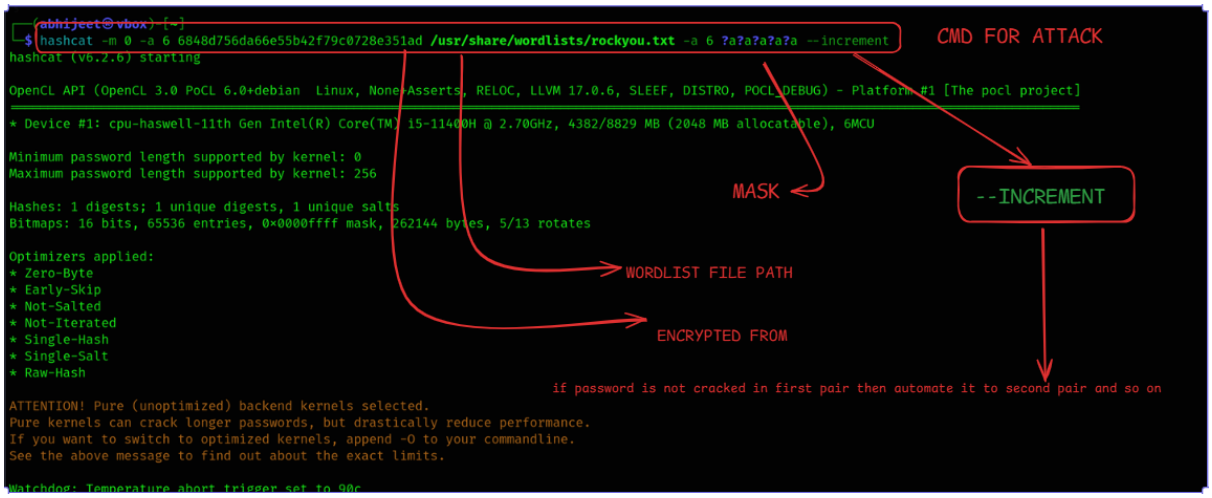
/usr/share/wordlists/rockyou.txt : This is a folder that contains 1.5cr passwords from all around the world installed in kali **.**

6848d756da66e55b42f79c0728e351ad : Encrypted from of password.

?a : represents all printable ASCII characters.

--increment: checks the password pair by pair and if not found increases the pair count by 1 and so on.

Example :



Result :

```
File  Actions  Edit  View  Help

* Append -w 3 to the commandline.
  This can cause your screen to lag.

* Update your backend API runtime / driver the right way:
  https://hashcat.net/faq/wrongdriver

* Create more work items to make use of your parallelization power:
  https://hashcat.net/faq/morework

6848d756da66e55b42f79c0728e351ad:baby        -> cracked passsword

Session..........: hashcat
Status...........: Cracked                   -> successful
Hash.Mode........: 0 (MD5)
Hash.Target......: 6848d756da66e55b42f79c0728e351ad   -> target encrypted password
Time.Started.....: Wed Feb 12 13:57:27 2025 (38 secs)
Time.Estimated...: Wed Feb 12 13:58:05 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt), Left Side
Guess.Mod........: Mask (?a) [1], Right Side
Guess.Queue.Base.: 1/1 (100.00%)
Guess.Queue.Mod..: 1/5 (20.00%)
Speed.#1.........: 24887.5 kH/s (10.92ms) @ Accel:512 Loops:95 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 936514560/1362716575 (68.72%)
Rejected.........: 0/936514560 (0.00%)
Restore.Point....: 9854976/14344385 (68.70%)
Restore.Sub.#1...: Salt:0 Amplifier:0-95 Iteration:0-95
Candidate.Engine.: Device Generator
Candidates.#1....: babbyphat20s -> ba3boy6
Hardware.Mon.#1..: Util: 90%

Started: Wed Feb 12 13:57:23 2025
```

E. HASHCAT HYBRID MASK + WORDLIST ATTACK MODE :

It is same as the **HYBRID WORDLIST + MASK ATTACK MODE** just inter changing the position of **mask** and **path of wordlist**

CMD :

“hashcat -m 0 -a 7 d76f3d05cc9ac98f1f9160274a39fe33 ?a?a?a?a?a /usr/share/wordlists/rockyou.txt –increment”

Where :

-m : type of hashcode

-a : attack mode

/usr/share/wordlists/rockyou.txt : This is a folder that contains 1.5cr passwords from all around the world installed in kali **.**

?a : represents all printable ASCII characters.

--increment: checks the password pair by pair and if not found increases the pair count by 1 and so on.

Example:

WEBSERVER PASWORD HACKING Tools

```
┌──(abhijeet㉿vbox)-[~]
└─$ hashcat -m 0 -a 7 202cb962ac59075b964b07152d234b70 ?a?a?a?a?a /usr/share/wordlists/rockyou.txt --increment
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================
* Device #1: cpu-haswell-11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz, 4382/8829 MB (2048 MB allocatable), 6MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Raw-Hash
```

CMD FOR ATTACK

WORDLIST FILE PATH

--INCREMENT

MASK

ENCRYPTED FROM

if password is not cracked in first pair then automate it to second pair and so on

Result:

```
File  Actions  Edit  View  Help
The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

202cb962ac59075b964b07152d234b70:123

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 202cb962ac59075b964b07152d234b70
Time.Started.....: Wed Feb 12 14:02:33 2025 (3 secs)
Time.Estimated...: Wed Feb 12 14:02:36 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt), Right Side
Guess.Mod........: Mask (?a) [1], Left Side
Guess.Queue.Base.: 1/1 (100.00%)
Guess.Queue.Mod..: 1/5 (20.00%)
Speed.#1.........: 23236.2 kH/s (3.55ms) @ Accel:32 Loops:1024 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 52433920/1362716575 (3.85%)
Rejected.........: 0/52433920 (0.00%)
Restore.Point....: 0/95 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:550912-551936 Iteration:0-1024
Candidate.Engine.: Device Generator
Candidates.#1....: s24992533 →  222202
Hardware.Mon.#1..: Util: 83%

Started: Wed Feb 12 14:02:20 2025
Stopped: Wed Feb 12 14:02:37 2025
```

cracked passsword

successful

target encrypted password

F.ASSOCIATION

It has a association file that contains some credentials related to password such as username , email etc.

CMD : “hashcat -m 0 -a 9 hash.txt asso.txt”

Where:

-m : type of hash -a : mode of attack

Hash.txt : encrypted file

Asso.txt : associated file that contains username , email , etc.

Example :

```
┌──(abhijeet㉿vbox)-[~/Desktop/HYDRA]
└─$ hashcat -m 0 -a 9 hash.txt aso.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================
* Device #1: cpu-haswell-11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz, 4382/8829 MB (2048 MB allocatable), 6MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
```

Association file

Result :

```
9c76e6ad775dafe6bae15b9cd6e6dba5:as@1

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 9c76e6ad775dafe6bae15b9cd6e6dba5
Time.Started.....: Tue Feb 18 12:13:39 2025 (0 secs)
Time.Estimated ...: Tue Feb 18 12:13:39 2025 (0 secs)
Kernel.Feature ...: Pure Kernel
Guess.Base.......: File (aso.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:      813 H/s (0.06ms) @ Accel:1 Loops:1 Thr:1 Vec:1
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 1/1 (100.00%)
Rejected.........: 0/1 (0.00%)
Restore.Point....: 0/1 (0.00%)
Restore.Sub.#1 ...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: as@1 → as@1
Hardware.Mon.#1..: Util: 15%

Started: Tue Feb 18 12:13:37 2025
Stopped: Tue Feb 18 12:13:41 2025
```

cracked successfully

2.HYDRA

Hydra is a brute-forcing tool that helps penetration testers and ethical hackers crack the passwords of network services.

A. Single Username/Password Attack with HYDRA :

If we have the username and password that we expect a system to have, we can use Hydra to test it.

CMD: “hydra -l ksqxmy0au8w1 -p AWkK6nMRiV%c 50.62.137.207 ssh”

Where :

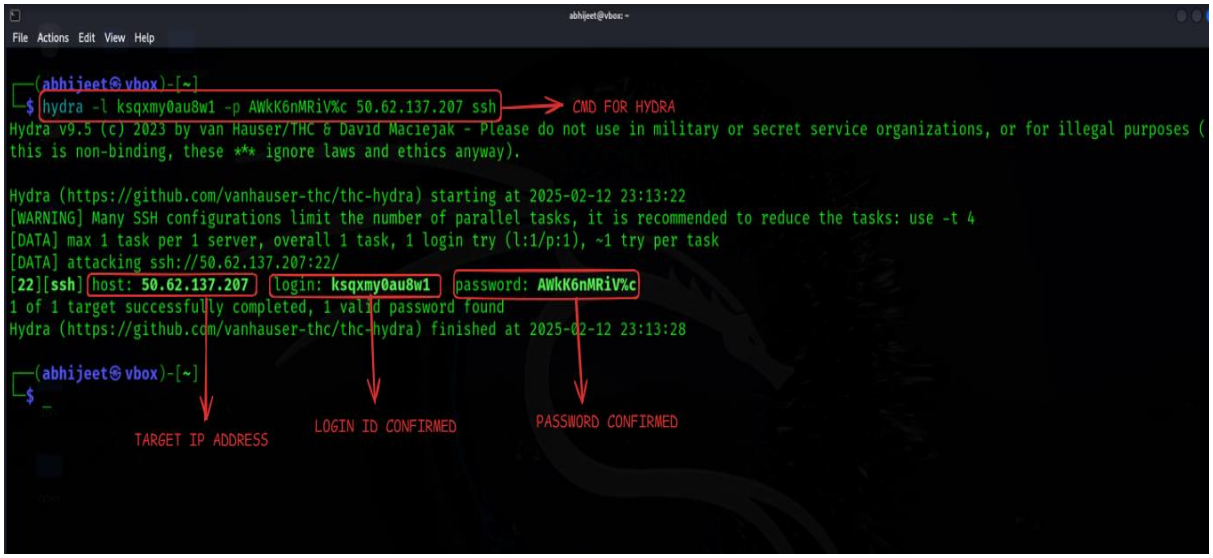
-l : login id

-p : password

50.62.137.207 : ip address of target

ssh : service

Example :



B. Password Spraying Attack using HYDRA :

If we have a confirmed password and have a file/list of username , then we can use this technique to confirm the username for the respective password.

CMD :

“hydra -L username.txt -p password <ip address> <service (ssh)>”

where :

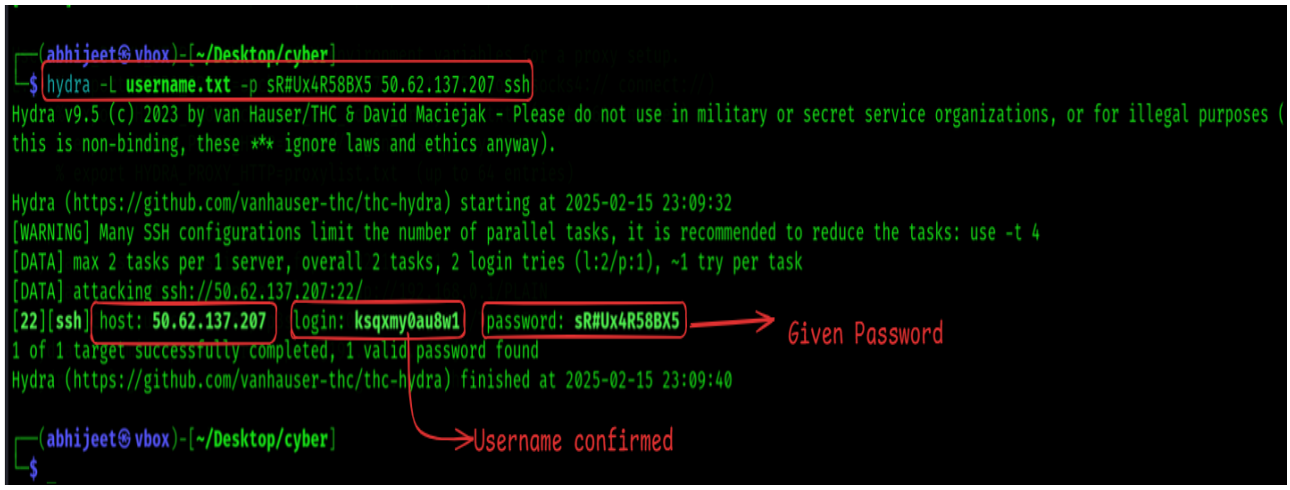
username.txt : all username list

-p : password

Ip : ip address of server

Service : ssh

Example :

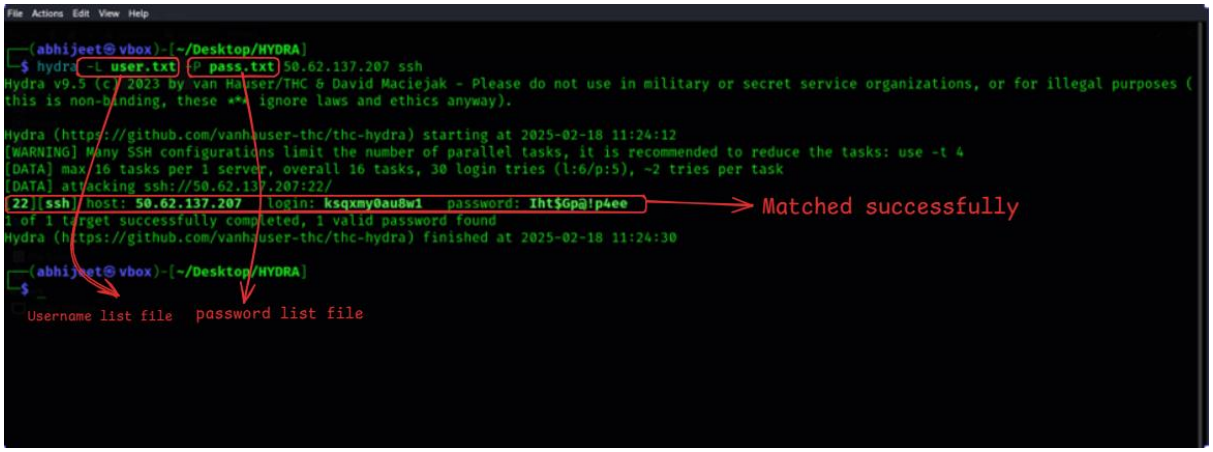


C . Dictionary Attack with HYDRA :

If we have multiple/single username and multiple/single passwords for a server .

We can use this to get the username and password for the respective server ip address.

CMD:"hydra -L user.txt -P pass.txt <server ip> ssh"

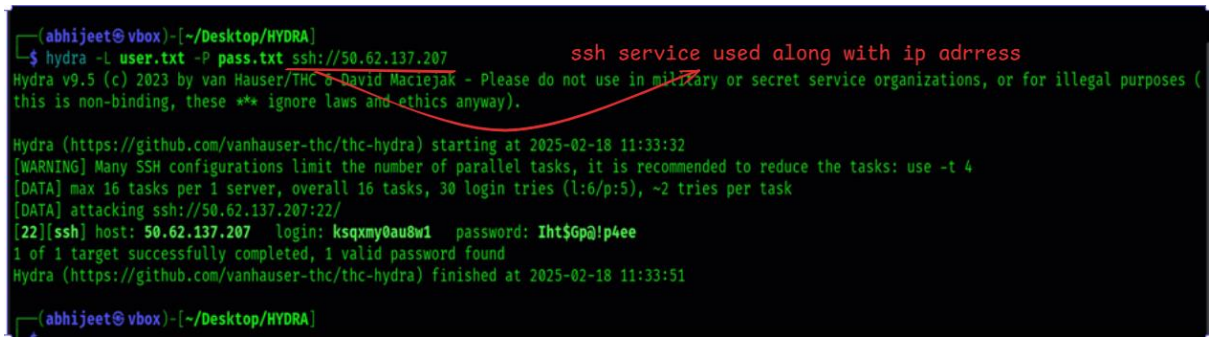


D . Various flags & format in HYDRA :

1.Service Specification :

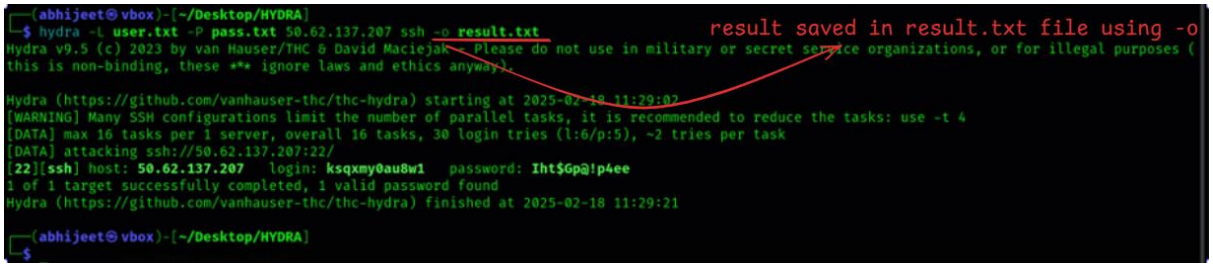
Instead of specifying service separately we can use it with respective ip address.

CMD : "hydra -L user.txt -P pass.txt ssh://<server ip>"

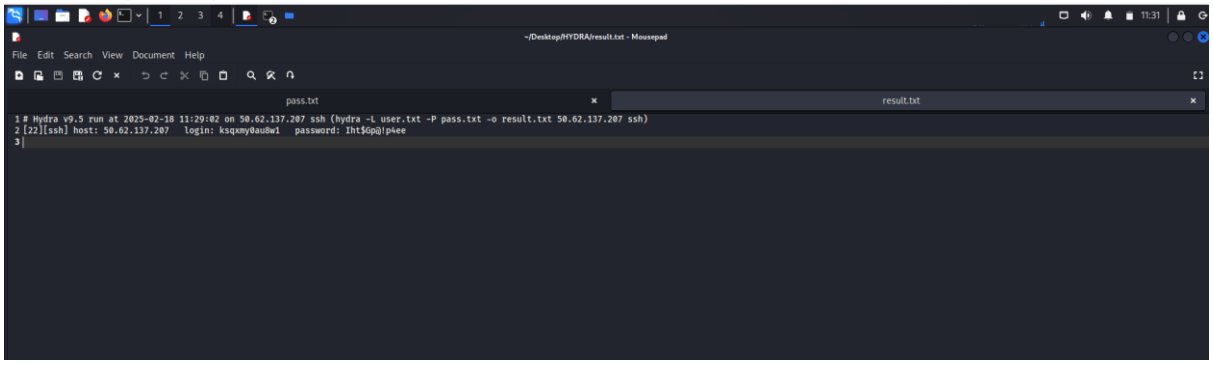


2.-o : it is used to store the result in the file

cmd: hydra -l username -p password <ip> ssh -o result.txt



Result saved :



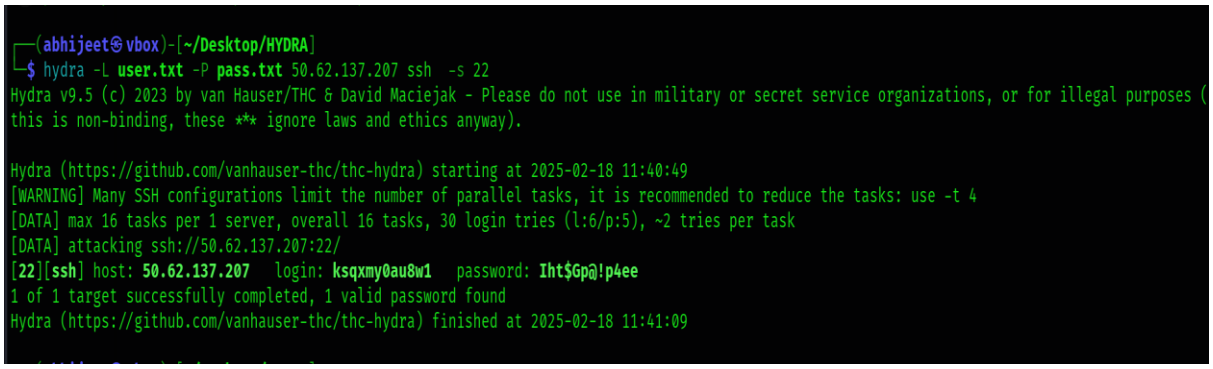
3.Custom ports :

We can target the server using custom ports

For now we will just look for ssh service that is running on port 22

.

CMD:"hydra -L user.txt -P pass.txt <server ip> ssh -s 22"



4.Attacking multiple ports:

This can be done in same way ,just we need to add a host file in which we will save the ip address of multiple hosts .

CMD: “hydra -L user.txt -P pass.txt -M hostfile.txt ssh ”

5.Targeted Combinations:

We provide a pair of combination of username and password .

It identifies the correct and match and provides the result .

CMD:”hydra -C combination.txt <ip> ssh”